

1 (1 punto) Describa la función de la política de ubicación en la jerarquía de memoria. Indique los tipos que se utilizan en memoria caché y sus posibles ventajas e inconvenientes desde el punto de vista del rendimiento.

SOLUCIÓN

Esta política decide dónde se puede almacenar un bloque de información (bloque de caché o página) en un determinado nivel de la jerarquía, y particularmente relevante en el caso de la caché.

En este caso, determina en qué línea o líneas puede albergarse un bloque que se sube de memoria principal. La política es más o menos *asociativa* dependiendo del número de líneas posibles entre las que se puede elegir, lo que decidiría entonces la política de reemplazo. La política *directa* (asociativa de grado 1) sólo permite almacenarlo en una línea determinada y resulta la más rápida, tanto por necesitar el menor número de bits (etiqueta o *tag*) más corto, como porque no es necesario invertir tiempo en la política de reemplazo. El extremo contrario se encuentra la política *completamente asociativa* en la que se puede albergar en cualquier línea, y que, por tanto, resulta la más lenta a la hora de su acceso, aunque podría dar lugar a una mejor tasa de aciertos. Como solución de compromiso se puede utilizar una política *asociativa por conjuntos*, en la que se puede elegir habitualmente entre 2 o 4 líneas, y que mejora la tasa de aciertos sin resultar tan lenta como la completamente asociativa.

2 (1 punto) Describa en qué consiste el fenómeno de proximidad de referencias (*locality*). Indique qué es la proximidad espacial y temporal, y explique, razonadamente, por qué tienden a darse tanto para direcciones de código como de datos. ¿En qué medida cree que este fenómeno es necesario para el adecuado funcionamiento de la jerarquía de memoria?

SOLUCIÓN

La estructura habitual de los programas hace que tienda a darse el hecho de que accedan repetidamente a las mismas direcciones o a direcciones cercanas, fenómeno que suele denominarse de proximidad de referencias (*locality*, en inglés). En general, suele distinguirse entre proximidad temporal –la traza del programa repite las mismas direcciones en instantes cercanos en el tiempo– y espacial –cuando la traza contiene direcciones próximas en el espacio de direcciones–, un subconjunto particular de la cual es la llamada proximidad secuencial –la traza contiene direcciones consecutivas o contiguas–.

En el caso de la proximidad temporal es fácil observar que tiende a darse en el código, principalmente por la existencia de bucles (o saltos “hacia atrás”), que vuelven a rerreferenciar direcciones de instrucciones, aunque también las correspondientes datos a las que este código hace referencia, o a posiciones cercanas. El hecho de que las direcciones de código tiendan a ser contiguas a partir de una bifurcación justifica la ocurrencia de la proximidad espacial para instrucciones, mientras que la de datos se da en la manera en que los compiladores almacenan las estructuras de información, en particular las de tipo vector o formación, aunque también lo suelen hacer en el caso de otras con apariencia menos “regular”, como las estructuras dinámicas o los espacios de almacenamiento temporal durante la ejecución, como puedan ser la pila y el *heap* o “montón”.

Precisamente el funcionamiento de la Jerarquía de Memoria (JM) se basa en la existencia de la proximidad de referencias, ya que los dispositivos más rápidos necesariamente tienen una capacidad menor, y contienen siempre la información más reciente o “fresca” y que volverá a ser referenciada. La JM no funcionaría si no se diese este fenómeno.

3 (1 punto) Sea una memoria caché de dos niveles, L1 –con cachés separadas para datos e instrucciones– y L2. Calcule razonadamente la tasa de aciertos global de esta caché supuestos los siguientes valores estadísticos:

- Tasa de aciertos de la memoria caché L1 de instrucciones: 94 %.
- Tasa de aciertos de la memoria caché L1 de datos: 91 %.
- El 30 % de los accesos es a datos.
- Tasa de aciertos local de la caché L2 es 38 %.

SOLUCIÓN

La tasa de aciertos global será la conjunta resultante del sistema de caché. Los accesos que no se satisfacen en el nivel L1 se tratan de satisfacer en el L2, donde la tasa de aciertos local es del 38 %. Como la tasa de aciertos del nivel L1 es diferente para datos y para instrucciones habrá que distinguir en tasa de aciertos global para cada tipo de acceso.

Tasa de aciertos global para datos:

$$Hr_{McaD} = Hr_{McaDL1} + (1 - Hr_{McaDL1}) \times Hr_{McaL2} = 0,91 + 0,09 \times 0,38 = 0,94$$

Tasa de aciertos global para instrucciones:

$$Hr_{McaI} = Hr_{McaIL1} + (1 - Hr_{McaIL1}) \times Hr_{McaL2} = 0,94 + 0,06 \times 0,38 = 0,96$$

Tasa de aciertos global media:

$$Hr_{Mca} = \%I \times Hr_{McaI} + \%D \times Hr_{McaD} = 0,7 \times 0,96 + 0,3 \times 0,94 = 0,954$$

4 (1 punto) Describa el funcionamiento de las tablas de página multinivel. Indique sus ventajas e inconvenientes frente al uso de un único nivel. Razone por qué se suele elegir un tamaño de tabla de páginas tal que cada una de las tablas de cada nivel ocupe una única página.

SOLUCIÓN

Las tablas de página multinivel pudieran tal vez definirse más significativamente como “tablas de página paginadas”, ya que la idea que subyace es tener en cada momento residente en memoria sólo los fragmentos necesarios de lo que sería tabla de página única (la que se obtendría con tantas entradas como paginas virtuales indexada directamente por PV). Para ello se suelen establecer varios niveles, de manera que el último sea el que contenga las PTEs con la traducción, de estar residentes, de la PV a MP y los demás niveles actúen como índices a estas tablas de páginas, con tablas de páginas que a su vez pueden estar o no residentes bajo demanda de la ejecución (sólo la única de primer nivel estará siempre residente).

La ventaja fundamental es el ahorro de memoria, ya que se cargarán bajo demanda sólo las tablas de páginas mínimas necesarias para el proceso según cada una de las zonas o regiones de memoria a las que acceda (y no siempre la tabla completa). El inconveniente es los potenciales fallos de tabla de página (que no se daría en el caso de la tabla única) y el tiempo que es necesario invertir en la sucesión de accesos a memoria necesaria para traducir a través de la cadena de TPs (hasta llegar al último nivel, que contiene la PTE en sí). En la vida real, sin embargo, esta sucesión de accesos sólo es necesaria en los fallos en la TLB, por lo que su coste no resulta muy significativo.

La gestión de la memoria principal que hace el S.O. es usando marcos de página, por lo que la manera más cómoda y lógica de ubicar las tablas de páginas es alojándola es uno solo de estos marcos. De esta forma permite la fácil indexación a partir de la dirección de comienzo del marco y evita tener que asignarles marcos de página obligatoriamente consecutivos en Mp.

5 (6 puntos) Sea un procesador con un tamaño de palabra de 32 bits y cuyo sistema de memoria tiene las siguientes características:

- Memoria virtual
 - Espacio virtual direccionable: 8 TB.
 - Tamaño de las páginas: 8 KB.
 - Niveles de tablas de páginas: 3.
 - Tamaño de las entradas de las tablas de páginas: 8 bytes (2 palabras).
 - Una TLB para datos y otra para instrucciones, con un tiempo de consulta de 1 ns.
- Memoria caché
 - Número de niveles: único, L1.
 - Memorias caché separadas para instrucciones y datos.
 - Capacidad de cada una de las cachés: 64 KB.

- Tamaño de los bloques: 64 bytes
 - Tiempo de acceso: 1 ns.
 - Política de ubicación: directa.
 - Política de lectura: out of order fetch
 - Política de escritura (caché de datos): aplazada (CBWA).
- En los fallos en escritura se modifica primero la Mp y se actualiza luego el bloque en la cache.

- Memoria principal

- Tiempo de acceso: 40 ns.
- Organización: entrelazado simple de orden inferior de 16 módulos.

a) Sea la dirección virtual $H'3FFFFFF000$. Indique razonadamente los campos en que se descompone esta dirección virtual y sus longitudes. Indique también el número de tablas de página de cada nivel que es necesario para su traducción.

b) Suponiendo que a la página que corresponde a la dirección anterior se le asigna el marco de página $H'0FFF$, describa los campos en que se divide la dirección física correspondiente tal y como es interpretada por las cachés. Indique además en qué línea o líneas se ubicaría en caché el bloque correspondiente a esta dirección.

c) Calcule los tiempos de traducción mínimo y máximo suponiendo que no se produce ningún fallo de página.

d) Calcule el tiempo mínimo de acceso al sistema de memoria, suponiendo de nuevo que no se produce ningún fallo de página. Indique razonadamente si se obtendrían diferentes valores para las lecturas y las escrituras.

e) Calcule el tiempo medio de acceso al sistema de memoria suponiendo los siguientes valores obtenidos en la ejecución de un programa:

- Tasa de aciertos de la memoria caché de instrucciones: 95 %.
- Tasa de aciertos de la memoria caché de datos: 93 %.
- Tasa de aciertos de ambas TLBs: 97 %.
- Todas las páginas referenciadas están residentes en memoria.
- El 80 % de los accesos corresponde a instrucciones. En los accesos a datos, el 70 % es de lectura.
- Probabilidad de reemplazar un bloque modificado: 30 %.

SOLUCIÓN

a) Los 8 TB significan que las direcciones virtuales tienen una longitud de 43 bits. Al ser las páginas de 8 KB, es decir 2^{13} bytes se necesitan 13 bits para identificar cualquier byte dentro de una página. Los 30 bits restantes de la dirección servirán para indexar los tres niveles de tablas de páginas. En este caso, la división para razonable sería dedicar 10 bits a cada nivel, ya que de esta forma cualquier tabla de páginas ocupa 1 página: 2^3 bytes (2 palabras) por cada entrada (PTE) y 2^{10} entradas por cada tabla de páginas: $2^{10} \times 2^3 = 2^{13}$ (1 página):

n0	n1	n2	byte
10 bits	10 bits	10 bits	13 bits

En el caso particular de la dirección $H'3FFFFFF000$, correspondería los tres índices apuntarían a la última entrada de las tablas de los tres niveles (todos los bits a 1). Lógicamente, se necesitaría una tabla de cada nivel para poder realizar la traducción.

b) Las direcciones físicas de 32 bits (con las que se accederá después de la traducción a la Mca) se interpretarán de la siguiente forma, de acuerdo a cómo está configurada la caché, y en particular a su política de ubicación:

En una memoria caché con política de ubicación directa, su controlador interpretará las direcciones físicas como formadas por tres campos: el byte en el bloque de caché, el bloque de caché o línea y la etiqueta. Para determinar el byte en el bloque hacen falta tantos bits como determine el tamaño de los bloques:

Bloques de 64 bytes: 2^6 bytes \rightarrow se necesitan 6 bits.

El número de bits que selecciona la línea vendrá dado por el número total de bloques de caché (c):

McaI y McaD de 64 KB: 2^{16} bytes $\rightarrow 2^{16}$ bytes / 2^6 bytes/bloque = 2^{10} bloques.

Es decir, se necesitan 10 bits para seleccionar la línea correspondiente a la dirección.

Finalmente, la etiqueta estará formada por el resto de los bits, hasta completar los correspondientes a la dirección física:

Con la dirección física de 32 bits: $(32 - 10 - 6)$ bits = 16 bits de etiqueta.

etiqueta	línea	byte
16 bits	10 bits	6 bits

En el caso particular de la dirección que se pide, H'3FFFFFFF000, los 30 bits de mayor peso (en este caso, todos a 1) indican la página virtual (PV) en que se encuentra. Una vez traducido el número de marco de página (MP) asignado, H'0FFF, concatenado con el desplazamiento de 13 bits H'1000: H'01FFF000. Consecuentemente los 10 bits que determinan la línea (única) en que se alojaría tienen el valor B'1111000000.

c) El tiempo mínimo de traducción corresponderá al acierto en TLB, mientras que el máximo se daría en el caso de que, por haber fallo en la TLB, hubiese que acceder a los tres niveles de tablas de página:

Tiempo mínimo de traducción:

$$t_{mtrad} = t_{TLB} = 1 \text{ ns}$$

El tiempo máximo de traducción es el que se da cuando hay fallo de TLB y, consecuentemente, se ha de acceder a los tres niveles de TPs, aunque el entrelazado –que permite leer las dos palabras de cada entrada en un tiempo de acceso a memoria– reduce el tiempo de fallo a la mitad, el equivalente a tres accesos a memoria:

$$t_{Mtrad} = t_{TLB} + t_{falloTLB} = t_{TLB} + 3 \times t_{Mp} = 1 \text{ ns} + 3 \times 40 \text{ ns} = 121 \text{ ns}$$

d) El tiempo mínimo será igual en el caso de direcciones correspondientes a instrucciones y a datos, y corresponderá al caso del tiempo mínimo de traducción que acabamos de calcular más (ya que no se puede realizar simultáneamente con traducción) el acceso a la caché. El tiempo máximo de acceso será la suma del tiempo máximo de traducción y el tiempo máximo de acceso a la información. Además, el tiempo máximo de traducción (acceso a 3 niveles de TPs –3 accesos a 2 palabras de Mp con entrelazado que permite obtener dos palabras consecutivas en un tiempo de acceso–, ya que se suponen todas residentes en memoria) será el mismo en los accesos a instrucciones y a datos. Sin embargo, el tiempo máximo de acceso será diferente según se trate de acceso a datos o a instrucciones, ya que en el acceso a datos el bloque que se reemplace pudiera estar modificado (*copy back*).

El tiempo mínimo en el acceso al sistema de memoria será igual en el caso de direcciones correspondientes a instrucciones y a datos, y corresponderá al tiempo mínimo de traducción (acierto en la TLB) y el acceso a la caché propiamente dicha. Por tanto:

tiempo mínimo (m), tanto para instrucciones como para datos:

$$t_m = t_{trad} + t_{Mca} = 1 \text{ ns} + 1 \text{ ns} = 2 \text{ ns}$$

El tiempo máximo de acceso a instrucciones, t_{MIns} , ocurrirá en el caso de fallo en caché y acceso a memoria para leer el bloque (teniendo en cuenta que se transmite primero la palabra a la que se hace referencia, al utilizarse *out of order fetch*). El tiempo de llegada de la primera palabra del bloque es igual al de lectura de una palabra, ya que no incluye el trasiego de las otras diecisiete del bloque:

$$t_{MIns} = t_{Mca} + t_{Mp} = 1 \text{ ns} + 40 \text{ ns} = 41 \text{ ns}$$

El tiempo máximo de acceso a datos, t_{MDat} , ocurrirá cuando se produzca fallo en caché en escritura y el bloque que se sustituye esté modificado, ya que en este caso, por ser la política de actualización CBWA, será necesario actualizar previamente en memoria principal el bloque “desalojado” de memoria caché. Nótese que en los movimientos de bloques se aprovecha el efecto del entrelazado, reduciendo este tiempo al de una palabra. También que, según el enunciado, se realiza primero la modificación en memoria antes de subir el bloque que ha provocado el fallo en escritura:

$$t_{MDat} = t_{Mca} + t_{Mp} + t_{bloque} + t_{Mp} = 1 \text{ ns} + 40 \text{ ns} + 40 \text{ ns} + 40 \text{ ns} = 121 \text{ ns}$$

Después de calculados los tiempos máximos de traducción y de acceso a instrucciones y a datos a partir de la dirección física, se obtienen los tiempos máximos de acceso siguientes, que incluyen la traducción (t_{MINST} para instrucciones, t_{MDAT} para datos):

$$t_{MINST} = 121 \text{ ns} + 41 \text{ ns} = 162 \text{ ns} \quad t_{MDAT} = 121 \text{ ns} + 121 \text{ ns} = 242 \text{ ns}$$

e) Análogamente al apartado anterior, los tiempos medios de acceso serán la suma del tiempo medio de traducción más el tiempo medio de acceso a la información, promediado en instrucciones y datos.

En el caso de acceso a datos, el bloque que se reemplaza puede estar modificado (lo que ocurre con una probabilidad P_{mod}) y en este caso se ha de bajar previamente en memoria. También los fallos en escritura (probabilidad P_E) añaden el tiempo de la escritura en M_p antes de subir la palabra y luego el resto del bloque.

$$\begin{aligned} \bar{t}_{AD} &= t_{Mca} + (1 - Hr_{McaD}) \times (t_{Mp} + P_{mod} \times t_{bloque} + t_{Mp} + P_E \times t_{Mp}) \\ &= 1 \text{ ns} + 0,07 \times (40 \text{ ns} + 0,30 \times 40 \text{ ns} + 0,30 \times 40 \text{ ns}) = 5,48 \text{ ns} \end{aligned}$$

Para las instrucciones, se repiten estos cálculos sin el término correspondiente a la posibilidad de que el bloque que se sustituye esté modificado ni a las escrituras, por lo que queda:

$$\begin{aligned} \bar{t}_{AI} &= t_{Mca} + (1 - Hr_{McaD}) \times t_{Mp} \\ &= 1 \text{ ns} + 0,05 \times 40 \text{ ns} = 3 \text{ ns} \end{aligned}$$

Para calcular el tiempo medio de traducción sabemos que nos encontramos directamente traducidas el 97 % de las direcciones, tanto de datos como de instrucciones. El caso de fallo, tal y como explicamos al calcular los tiempos máximos, será necesario visitar los tres niveles de TPs:

$$\begin{aligned} \bar{t}_{trad} &= t_{TLB} + (1 - Hr_{TLB}) t_{falloTLB} = t_{TLB} + (1 - Hr_{TLB}) \times 3 \times t_{Mp} = \\ &= 1 \text{ ns} + 0,03 \times 3 \times 40 \text{ ns} = 3,6 \text{ ns} \end{aligned}$$

Finalmente, agregando el tiempo medio de traducción a los de accesos a la información, se obtienen los siguientes tiempos medios de acceso para instrucciones y datos:

$$\bar{t}_{accI} = 3,6 \text{ ns} + 3 \text{ ns} = 6,6 \text{ ns}$$

$$\bar{t}_{accD} = 3,6 \text{ ns} + 5,48 \text{ ns} = 9,08 \text{ ns}$$

1 [0,75 puntos] *Describa razonadamente cómo influye el grado de asociatividad de la memoria caché en: 1) su tasa de aciertos; y 2) su tiempo de acceso.*

SOLUCIÓN

Se comprueba experimentalmente que, en general, cuanto más asociativa es una caché, mayor es la tasa de aciertos que se obtiene. En tal medida es así que se ha observado que en las ejecuciones en un caché de tamaño C y grado de asociatividad 1 (directa) se obtiene prácticamente la misma tasa de aciertos que en una caché de la mitad de capacidad ($C/2$) y el doble de asociatividad (asociativa de grado 2 o con dos líneas por conjunto).

En el caso del tiempo de acceso, es una cuestión más de naturaleza tecnológica —pudiera tener que ver con la manera en se realiza la comprobación asociativa de las etiquetas y cómo se ve afectada por la longitud de éstas—, pero se comprueba que, en la vida real, las cachés con política de ubicación directa (asociatividad de grado 1) o con sólo dos líneas por conjunto son las que se utilizan habitualmente en las cachés L1 porque obtienen el tiempo menor en los aciertos.

2 [0,75 puntos] *Indique por qué el tamaño de los bloques de caché es del orden de unos cuantos bytes (habitualmente, 32 ó 64 bytes) mientras que el de las páginas es al menos del orden de KB (kilobytes).*

SOLUCIÓN

El tamaño de los bloques de caché suele ser del orden de “unos cuantos bytes”, normalmente coincidiendo o siendo múltiplo del entrelazado de la memoria principal (32 ó 64 bytes son tamaños típicos). Este tamaño actúa en favor de la proximidad espacial, a la vez que posibilita el hecho de que la penalización ante los fallos de caché sea mínima.

A diferencia de los bloques de caché, las páginas se eligen de una longitud que permita aprovechar la operación de E/S a disco a la que dan lugar y que suele ser del orden de KB. La naturaleza de este tipo de operaciones implica, a diferencia de los fallos de caché, la intervención del S.O. y justifica un cambio de contexto y la consiguiente replanificación del sistema.

3 [0,75 puntos] *Indique las posibles ventajas e inconvenientes de tratar por hardware o por software los fallos en la TLB.*

SOLUCIÓN

Modernamente parece que triunfa la gestión por software de los fallos de TLB. Ello implica la necesidad de que la arquitectura del procesador suministre una excepción o *trap* similar a la que se emplea para los fallos de página. Este funcionamiento tiene la ventaja de la flexibilidad en cuanto a la estructura de tablas de página y de sus entradas (PTE), que en el caso del tratamiento por hardware está predeterminada (aunque permita siempre un cierto grado de “programabilidad”), y que obliga en cierta forma al S.O. a asumir. Esta solución software tiene, sin embargo, el inconveniente del sobre coste que significa la ejecución de código cada vez que se produce un fallo de este tipo, aunque se suele conseguir rutinas con un reducido número de instrucciones.

4 [0,75 puntos] *Describa el funcionamiento de las tablas de página multinivel. Indique sus ventajas e inconvenientes frente al uso de un único nivel. Razone por qué se suele elegir un tamaño de tabla de páginas tal que cada una de las tablas de cada nivel ocupe una única página.*

SOLUCIÓN

Las tablas de página multinivel pudieran tal vez definirse más significativamente como “tablas de página paginadas”, ya que la idea que subyace es tener en cada momento residente en memoria sólo los fragmentos necesarios de lo que sería tabla de página única (la que se obtendría con tantas entradas como páginas virtuales indexada directamente por PV). Para ello se suelen establecer varios niveles, de manera que el último sea el

que contenga las PTEs con la traducción, de estar residentes, de la PV a MP y los demás niveles actúen como índices a estas tablas de páginas, con tablas de páginas que a su vez pueden estar o no residentes bajo demanda de la ejecución (sólo la única de primer nivel estará siempre residente).

La ventaja fundamental es el ahorro de memoria, ya que se cargarán bajo demanda sólo las tablas de páginas mínimas necesarias para el proceso según cada una de las zonas o regiones de memoria a las que acceda (y no siempre la tabla completa). El inconveniente es los potenciales fallos de tabla de página (que no se daría en el caso de la tabla única) y el tiempo que es necesario invertir en la sucesión de accesos a memoria necesaria para traducir a través de la cadena de TPs (hasta llegar al último nivel, que contiene la PTE en sí). En la vida real, sin embargo, esta sucesión de accesos sólo es necesaria en los fallos en la TLB, por lo que su coste no resulta muy significativo.

La gestión de la memoria principal que hace el S.O. es usando marcos de página, por lo que la manera más cómoda y lógica de ubicar las tablas de páginas es alojándola es uno solo de estos marcos. De esta forma permite la fácil indexación a partir de la dirección de comienzo del marco y evita tener que asignarles marcos de página obligatoriamente consecutivos en Mp.

5 [7 puntos] Sea un procesador con un tamaño de palabra de 32 bits y cuyo sistema de memoria tiene las siguientes características:

- Memoria virtual

- Espacio virtual direccionable: 4 TB.
- Tamaño de las páginas: 4 KB.
- Niveles de tablas de páginas: 3.
- Tamaño de las entradas de las tablas de páginas: 4 bytes (1 palabra).
- Una TLB para datos y otra para instrucciones, con un tiempo de acceso de 1 ns.

- Memoria caché

- Número de niveles: único, L1.
- Cachés separadas para instrucciones y datos.
- Capacidad de cada una de las cachés: 32 KB.
- Tamaño de los bloques: 32 bytes
- Tiempo de acceso: 1 ns.
- Política de ubicación: directa.
- Política de lectura: out of order fetch
- Política de escritura (caché de datos): aplazada (CBWA).

En los fallos en escritura se modifica primero la Mp y se actualiza luego el bloque en la caché.

- Memoria principal

- Espacio físico direccionable: 4 GB.
- Tiempo de acceso: 40 ns.
- Organización: entrelazado simple de orden inferior de 16 módulos.

En un intervalo de tiempo dado, un proceso accede a las siguientes tres zonas o regiones de su espacio de direcciones (se muestra su dirección de comienzo, su tamaño y si su recorrido es hacia direcciones crecientes o decrecientes):

- Zona 0: H'000000000000; 65 KB; crecientes
- Zona 1: H'010000000000; 6 KB; crecientes
- Zona 2: H'3FFFFFFF; 2 KB; decrecientes

a) Para cada una de las tres zonas:

1. Identifique las páginas virtuales que ocupa cada zona.
2. Indique el número e identificación de las tablas de páginas de cada uno de los tres niveles necesarias para la traducción de cada zona.

b) Suponga que el S.O. ha asignado a la primera página virtual de la zona 1 el marco de página H'00FFF. Indique razonadamente:

1. El contenido relevante de las entradas (PTEs) de los tres niveles de tablas de página necesarios para la traducción de esa primera página de la zona 1. Para ello, elija arbitrariamente las direcciones de memoria que estime necesarias.
2. El contenido relevante de la entrada de la TLB que contuviese la traducción de esta página.
3. Los tiempos mínimo y máximo de traducción para las direcciones contenidas en esta página y las circunstancias en que éstos se darían.
4. La línea o líneas en que se podrá alojar en caché el bloque que contiene la dirección de comienzo de la zona.
5. Los tiempos mínimo y máximo de acceso para esta dirección de comienzo suponiendo que corresponde a una región de datos.

SOLUCIÓN

a) En primer lugar, descompongamos las direcciones virtuales (42 bits) en los campos página virtual (PV) y desplazamiento dentro de la página (12 bits, dado su tamaño de 4 KB, 2^{12} bytes) :

PV	desplazamiento
30 bits	12 bits

Consecuentemente, para cada una de las zonas o regiones se obtiene que ocupan las siguientes páginas:

- Zona 0: 17 páginas consecutivas a partir de la página H'00000000.
- Zona 1: 2 páginas consecutivas a partir de la página H'01000000.
- Zona 2: 1 página, H'3FFFFFFF

Para identificar las tablas de páginas necesarias debemos primero determinar la estructura que se establece. El campo PV tiene 30 bits que es necesario dividir en tres índices, tal y como establece el enunciado. Una división probable es de manera uniforme, con 10 bits por nivel. Comprobamos que así se conseguiría que cada tabla de páginas ocupase una página (ya que cada entrada o PTE es de una palabra o 4 bytes):

$$2^{10} \text{ entradas} \rightarrow 2^{10} \times 2^2 \text{ bytes/entradas} = 2^{12} \text{ bytes} \rightarrow 1 \text{ página.}$$

n0	n1	n2	desplazamiento
10 bits	10 bits	10 bits	12 bits

Según esta estructura, el campo PV se descompone en tres índices, uno por cada nivel, quedando así por tanto identificadas las tablas de cada nivel necesarias:

- Zona 0: 16 páginas desde H'00000000 \rightarrow TPn0 [única], TPn1"H'000" [una TPn1], TPn2"H'000" [una TPn2]
- Zona 1: 2 páginas desde H'01000000 \rightarrow TPn0 [única], TPn1"H'010" [una TPn1], TPn2"H'000" [una TPn2]
- Zona 2: 1 página, H'3FFFFFFF \rightarrow TPn0 [única], TPn1"H'3FF" [una TPn1], TPn2"H'3FF" [una TPn2]

b) Si suponemos el S.O. ha asignado a la primera página virtual de la zona 1 el marco de página H'00FFF:

1. Habrá tres tablas de página involucradas. En la TPn0, la entrada número H'010 tendrá el bit de residencia a 1 y la dirección que suponemos "a" de comienzo de la TPn2 correspondiente. La entrada H'000 (siguientes 10 bits) tendrá también su bit de residencia a 1 y contendrá la dirección que llamaremos "b" de comienzo de la TPn2 correspondiente a ese índice. Finalmente, la entrada H'000 (10 bits) de esa tabla contendrá la PTE propiamente dicha, con la traducción al marco de página (MP), H'00FFF, y todos los demás bits de control, entre ellos el bit de residencia a 1.

2. La entrada correspondiente de la TLB tendrá el campo PV (por el que se busca asociativamente la traducción) al valor de la página, 30 bits, i.e., H'01000000, el valor del MP en que está albergada y todos los demás bits de control de la PTE.

3. El tiempo mínimo de traducción corresponderá al acierto en TLB, mientras que el máximo se daría en el caso de que, por haber fallo en la TLB, hubiese que acceder a los tres niveles de tablas de página:

Tiempo mínimo de traducción:

$$t_{mtrad} = t_{TLB} = 1 \text{ ns}$$

El tiempo máximo de traducción es el que se da cuando hay fallo de TLB y, consecuentemente, se ha de acceder a los tres niveles de TPs:

$$t_{Mtrad} = t_{TLB} + t_{falloTLB} = t_{TLB} + 3 \times t_{Mp} = 1 \text{ ns} + 3 \times 40 \text{ ns} = 121 \text{ ns}$$

4. Las direcciones físicas de 32 bits (con las que se accederá después de la traducción a la Mca) se interpretarán de la siguiente forma, de acuerdo a cómo está configurada la caché, y en particular a su la política de ubicación:

En una memoria caché con política de ubicación directa, su controlador interpretará las direcciones físicas como formadas por tres campos: el byte en el bloque de caché, el bloque de caché o línea y la etiqueta. Para determinar el byte en el bloque hacen falta tantos bits como determine el tamaño de los bloques:

Bloques de 32 bytes: 2^5 bytes \rightarrow se necesitan 5 bits.

El número de bits que selecciona la línea vendrá dado por el número total de bloques de caché (c):

McaI y McaD de 32 KB: 2^{15} bytes $\rightarrow 2^{15}$ bytes / 2^5 bytes/bloque = 2^{10} bloques.

Es decir, se necesitan 10 bits para seleccionar la línea correspondiente a la dirección.

Finalmente, la etiqueta estará formada por el resto de los bits, hasta completar los correspondientes a la dirección física:

Con la dirección física de 32 bits: $(32 - 10 - 5)$ bits = 17 bits de etiqueta.

etiqueta	línea	byte
17 bits	10 bits	5 bits

La primera dirección de la página será H'00FFF concatenado con el desplazamiento de 12 bits H'000: H'00FFF000. Consecuentemente los 10 bits que determina la línea (única) en que se alojaría es B'1110000000.

5. El tiempo mínimo en el acceso al sistema de memoria será igual en el caso de direcciones correspondientes a instrucciones y a datos, y corresponderá al tiempo mínimo de traducción (acierto en la TLB) y el acceso a la caché propiamente dicha.

Tiempo mínimo (m):

$$t_m = t_{trad} + t_{Mca} = 1 \text{ ns} + 1 \text{ ns} = 2 \text{ ns}$$

El tiempo máximo de acceso a datos, t_{Mdat} , ocurrirá cuando se produzca fallo en caché en escritura y el bloque que se sustituye esté modificado, ya que en este caso, por ser la política de actualización CBWA, será necesario actualizar previamente en memoria principal el bloque "desalojado" de memoria caché. Nótese que en los movimientos de bloques se aprovecha el efecto del entrelazado, reduciendo este tiempo al de una palabra. También que, según el enunciado, se realiza primero la modificación en memoria antes de subir el bloque que ha provocado el fallo en escritura:

$$t_{MDat} = t_{Mca} + t_{Mp} + t_{WRbloque} + t_{RDbloque} = 1 \text{ ns} + 40 \text{ ns} + 40 \text{ ns} + 40 \text{ ns} = 121 \text{ ns}$$

El tiempo máximo de acceso sería por tanto el siguiente:

$$t_{MDAT} = 121 \text{ ns} + 121 \text{ ns} = 242 \text{ ns}$$

1 [1,5 puntos] Describa la función de la política de ubicación en la jerarquía de memoria. Indique los tipos que se utilizan en memoria caché y sus posibles ventajas e inconvenientes desde el punto de vista del rendimiento.

SOLUCIÓN

Esta política decide dónde se puede almacenar un bloque de información (bloque de caché o página) en un determinado nivel de la jerarquía, y particularmente relevante en el caso de la caché.

En este caso, determina en qué línea o líneas puede albergarse un bloque que se sube de memoria principal. La política es más o menos *asociativa* dependiendo del número de líneas posibles entre las que se puede elegir, lo que decidiría entonces la política de reemplazo. La política *directa* (asociativa de grado 1) sólo permite almacenarlo en una línea determinada y resulta la más rápida, tanto por necesitar el menor número de bits (etiqueta o *tag*) más corto, como porque no es necesario invertir tiempo en la política de reemplazo. El extremo contrario se encuentra la política *completamente asociativa* en la que se puede albergar en cualquier línea, y que, por tanto, resulta la más lenta a la hora de su acceso, aunque podría dar lugar a una mejor tasa de aciertos. Como solución de compromiso se puede utilizar una política *asociativa por conjuntos*, en la que se puede elegir habitualmente entre 2 o 4 líneas, y que mejora la tasa de aciertos sin resultar tan lenta como la completamente asociativa.

2 [1,5 puntos] Responda razonadamente si son ciertas las siguientes afirmaciones sobre la memoria virtual:

- a) La memoria virtual consigue que reduzca el tiempo medio de acceso o tiempo efectivo.
- b) El tamaño de las páginas (y de los marcos de página) suele ser del orden de unos cuantos KB porque el trasiego de éstas con la memoria secundaria implica una operación de Entrada/Salida.
- c) Los fallos de página deben de tratarse por software del Sistema Operativo, por lo que las arquitecturas suelen suministrar una excepción o *trap* de fallo de página.
- d) La TLB reduce el tiempo mínimo de traducción a un valor igual al número de niveles de tabla de página (n) multiplicado por el tiempo de TLB (t_{TLB}), i.e., $n \times t_{TLB}$

SOLUCIÓN

- a) Es una afirmación falsa, puesto que la memoria virtual conlleva al menos el tiempo extra de traducción, que es al menos el tiempo de consulta de la TLB, y no digamos en el caso de que se produjese un fallo de página y todo el proceso al que éste da lugar.
- b) Es cierto. A diferencia de los bloques de caché, que tienen un tamaño de unos cuantos bytes, las páginas se eligen de una longitud que permita aprovechar la operación de ES a disco a la que dan lugar y que suele ser del orden de KB.
- c) Se trata también de un aserto correcto, puesto que resulta imprescindible la intervención del S.O para encargar la operación de E/S correspondiente y demás acciones que esta circunstancia conlleva (actualizar las diferentes colas, dar paso a otro proceso, etc.)
- d) Es otra afirmación incorrecta. La TLB contiene pares página virtual-marco de página, PV-MP, y su tiempo de consulta cuando se encuentra la traducción es independiente del número de niveles de tablas de página que se emplee, es decir, siempre t_{TLB}

3 Sea un procesador con un tamaño de palabra de 32 bits y cuyo sistema de memoria tiene las siguientes características:

- Memoria virtual
 - Espacio virtual direccionable: 8 TB.
 - Tamaño de las páginas: 8 KB.
 - Niveles de tablas de páginas: 3.
 - Tamaño de las entradas de las tablas de páginas: 8 bytes (2 palabras).
 - Una TLB para datos y otra para instrucciones, con un tiempo de consulta de 1 ns.
- Memoria caché

- Número de niveles: único, L1.
- Cachés separadas para instrucciones y datos.
- Capacidad de cada una de las cachés: 64 KB.
- Tamaño de los bloques: 64 bytes
- Tiempo de acceso: 1 ns.
- Política de ubicación: directa.
- Política de lectura: *out of order fetch*
- Política de escritura (caché de datos): aplazada (*CBWA*).

En los fallos en escritura se modifica primero la Mp y se actualiza luego el bloque en la caché.

- Memoria principal

- Tiempo de acceso: 40 ns.
- Organización: entrelazado simple de orden inferior de 16 módulos.

- a) Indique por qué se utilizan habitualmente cachés y TLBs separadas para instrucciones y datos.
- b) En un instante determinado durante su ejecución un programa ocupa tres áreas diferentes de memoria, Z1, Z2 y Z3, cuyo tamaño respectivo es:

- Z1: 32.272.423 bytes.
- Z2: 15.989.456 bytes.
- Z3: 2.342 bytes.

b.1) Calcule el número máximo de tablas de página de cada uno de los niveles que se necesitará para el conjunto de las tres áreas ocupadas por ese proceso en ese instante.

b.2) Para este mismo caso calcule ahora el número mínimo de tablas de página que sería necesario.

c) Calcule el tiempo de traducción mínimo y máximo suponiendo que no se produce ningún fallo de página.

d) Calcule el tiempo de acceso mínimo al sistema de memoria, suponiendo de nuevo que no se produce ningún fallo de página. Indique razonadamente si se obtendrían diferentes valores para las lecturas y las escrituras.

e) Calcule el tiempo medio de acceso al sistema de memoria suponiendo los siguientes valores obtenidos en la ejecución de un programa:

- Tasa de aciertos de la memoria caché de instrucciones: 95 %.
- Tasa de aciertos de la memoria caché de datos: 93 %.
- Tasa de aciertos de ambas TLBs: 97 %.
- Todas las páginas referenciadas están residentes en memoria.
- El 80 % de los accesos corresponde a instrucciones. En los accesos a datos, el 70 % es de lectura.
- Probabilidad de reemplazar un bloque modificado: 30 %.

SOLUCIÓN

a) Las cachés desdobladas en el nivel más alto de la jerarquía permiten el acceso en el mismo ciclo de reloj a instrucciones (*fetch*) y datos, lo que resulta imprescindible en una estructura con *pipeline*, en la que sería necesario introducir parones cada vez que concurriese esta circunstancia.

Si la jerarquía dispone de un mecanismo de memoria virtual resultará necesario que las TLB se encuentren también replicadas para permitir la traducción en paralelo de las direcciones virtuales de instrucciones y datos, lo que de nuevo evitará la necesidad de introducir parones en los instantes en se acceda a la vez a direcciones de instrucciones y de datos.

b) Con la estructura de tabla de páginas descrita en el apartado anterior, se conforman regiones de 8 MB (2^{10} páginas \times 8 KB/página) agrupadas en zonas que contienen a su vez 8 GB (2^{10} regiones \times 8 MB/región)

Para cada una de las tres áreas se obtendría por tanto:

- Z1: 32.272.423 bytes \rightarrow 30'78 MB \rightarrow 3'85 regiones \rightarrow 4 regiones \rightarrow 1 zona (2 zonas si no "alineada a zona") (5 regiones si no "alineada a región")
- Z2: 15.989.456 bytes \rightarrow 15'25 MB \rightarrow 1,9 regiones \rightarrow 2 regiones (3 regiones si no "alineada a región") \rightarrow 1 zona (2 zonas si no "alineada a zona")

- Z3: 2.342 bytes \rightarrow 1 página (2 páginas, si no está “alineada a página”) \rightarrow 1 región (2 regiones si no “alineada a región”) \rightarrow 1 zona (2 zonas si no “alineada a zona”)

b.1) El máximo número de tablas que ocuparían estas tres áreas se daría en el caso de que estuviese lo suficientemente “dispersas” en el espacio de direcciones del proceso para ocupar zonas diferentes y empleando por tanto tantas tablas de páginas de nivel 2 (TP2), las que “apuntan” a las regiones, como zonas diferentes. Además siempre habrá que tener en cuenta la posibilidad de que el área no comience al principio de una región, sino que no esté “alineada”, por lo que ocupará una región más, tal y como indicamos en el desglose anterior. La misma consideración se puede hacer con las zonas de 8 TB. También hay que tener en cuenta que la única tabla de página de nivel 1 (TP1), la que “apunta” a las áreas, estará siempre residente. En consecuencia:

- Z1: TP1; 2 TP2s; 5 TP3s
- Z2: TP1; 2 TP2s; 3 TP3s
- Z3: TP1; 2 TP2s; 2 TP3s

En total, por tanto, se emplearán TP1, 6 TP2, 10 TP3.

b.2) En el caso del mínimo los cálculos se simplifican, ya que, en contraposición al caso anterior, las áreas estarán lo menos dispersas y “alineadas” a región y a zona:

Total de memoria utilizada: 48.264.221 bytes \rightarrow 46'03 MB \rightarrow 5'75 regiones \rightarrow 6 regiones \rightarrow 1 zona

y el número total de tablas de páginas de cada nivel será: TP1, 1 TP2, 6 TP3

c) El tiempo mínimo de traducción corresponderá al acierto en TLB, mientras que el máximo se daría en el caso de que, por haber fallo en la TLB, hubiese que acceder a los tres niveles de tablas de página:

Tiempo mínimo de traducción:

$$t_{mtrad} = t_{TLB} = 1 \text{ ns}$$

El tiempo máximo de traducción es el que se da cuando hay fallo de TLB y, consecuentemente, se ha de acceder a los tres niveles de TPs, aunque el entrelazado –que permite leer las dos palabras de cada entrada en un tiempo de acceso a memoria– reduce el tiempo de fallo a la mitad, el equivalente a tres accesos a memoria:

$$t_{Mtrad} = t_{TLB} + t_{falloTLB} = t_{TLB} + 3 \times t_{Mp} = 1 \text{ ns} + 3 \times 40 \text{ ns} = 121 \text{ ns}$$

d) El tiempo mínimo será igual en el caso de direcciones correspondientes a instrucciones y a datos, y corresponderá al caso del tiempo mínimo de traducción que acabamos de calcular más (ya que no se puede realizar simultáneamente con traducción) el acceso a la caché. El tiempo máximo de acceso será la suma del tiempo máximo de traducción y el tiempo máximo de acceso a la información. Además, el tiempo máximo de traducción (acceso a 3 niveles de TPs –3 accesos a 2 palabras de Mp con entrelazado que permite obtener dos palabras consecutivas en un tiempo de acceso–, ya que se suponen todas residentes en memoria) será el mismo en los accesos a instrucciones y a datos. Sin embargo, el tiempo máximo de acceso será diferente según se trate de acceso a datos o a instrucciones, ya que en el acceso a datos el bloque que se reemplaza pudiera estar modificado (*copy back*).

El tiempo mínimo en el acceso al sistema de memoria será igual en el caso de direcciones correspondientes a instrucciones y a datos, y corresponderá al tiempo mínimo de traducción (acierto en la TLB) y el acceso a la caché propiamente dicha. Por tanto:

tiempo mínimo (m), tanto para instrucciones como para datos:

$$t_m = t_{trad} + t_{Mca} = 1 \text{ ns} + 1 \text{ ns} = 2 \text{ ns}$$

El tiempo máximo de acceso a instrucciones, t_{MIns} , ocurrirá en el caso de fallo en caché y acceso a memoria para leer el bloque (teniendo en cuenta que se transmite primero la palabra a la que se hace referencia, al utilizarse *out of order fetch*). El tiempo de llegada de la primera palabra del bloque es igual al de lectura de una palabra, ya que no incluye el trasiego de las otras diecisiete del bloque:

$$t_{MIns} = t_{Mca} + t_{Mp} = 1 \text{ ns} + 40 \text{ ns} = 41 \text{ ns}$$

El tiempo máximo de acceso a datos, t_{Mdat} , ocurrirá cuando se produzca fallo en caché en escritura y el bloque que se sustituye esté modificado, ya que en este caso, por ser la política de actualización CBWA, será necesario actualizar previamente en memoria principal el bloque “desalojado” de memoria caché. Nótese que en los movimientos de bloques se aprovecha el efecto del entrelazado, reduciendo este tiempo al de una palabra. También que, según el enunciado, se realiza primero la modificación en memoria antes de subir el bloque que ha provocado el fallo en escritura:

$$t_{MDat} = t_{Mca} + t_{Mp} + t_{bloque} + t_{Mp} = 1 \text{ ns} + 40 \text{ ns} + 40 \text{ ns} + 40 \text{ ns} = 121 \text{ ns}$$

Después de calculados los tiempos máximos de traducción y de acceso a instrucciones y a datos a partir de la dirección física, se obtienen los tiempos máximos de acceso siguientes, que incluyen la traducción (t_{MINST} para instrucciones, t_{MDAT} para datos):

$$t_{MINST} = 121 \text{ ns} + 41 \text{ ns} = 162 \text{ ns} \quad t_{MDAT} = 121 \text{ ns} + 121 \text{ ns} = 242 \text{ ns}$$

e) Análogamente al apartado anterior, los tiempos medios de acceso serán la suma del tiempo medio de traducción más el tiempo medio de acceso a la información, promediado en instrucciones y datos.

En el caso de acceso a datos, el bloque que se reemplaza puede estar modificado (lo que ocurre con una probabilidad P_{mod}) y en este caso se ha de bajar previamente en memoria. También los fallos en escritura (probabilidad P_E) añaden el tiempo de la escritura en Mp antes de subir la palabra y luego el resto del bloque.

$$\begin{aligned} \bar{t}_{AD} &= t_{Mca} + (1 - Hr_{McaD}) \times (t_{Mp} + P_{mod} \times t_{bloque} + t_{Mp} + P_E \times t_{Mp}) \\ &= 1 \text{ ns} + 0,07 \times (40 \text{ ns} + 0,30 \times 40 \text{ ns} + 0,30 \times 40 \text{ ns}) = 5,48 \text{ ns} \end{aligned}$$

Para las instrucciones, se repiten estos cálculos sin el término correspondiente a la posibilidad de que el bloque que se sustituye esté modificado ni a las escrituras, por lo que queda:

$$\begin{aligned} \bar{t}_{AI} &= t_{Mca} + (1 - Hr_{McaD}) \times t_{Mp} \\ &= 1 \text{ ns} + 0,05 \times 40 \text{ ns} = 3 \text{ ns} \end{aligned}$$

Para calcular el tiempo medio de traducción sabemos que nos encontramos directamente traducidas el 97 % de las direcciones, tanto de datos como de instrucciones. El caso de fallo, tal y como explicamos al calcular los tiempos máximos, será necesario visitar los tres niveles de TPs:

$$\begin{aligned} \bar{t}_{trad} &= t_{TLB} + (1 - Hr_{TLB}) t_{falloTLB} = t_{TLB} + (1 - Hr_{TLB}) \times 3 \times t_{Mp} = \\ &= 1 \text{ ns} + 0,03 \times 3 \times 40 \text{ ns} = 3,6 \text{ ns} \end{aligned}$$

Finalmente, agregando el tiempo medio de traducción a los de accesos a la información, se obtienen los siguientes tiempos medios de acceso para instrucciones y datos:

$$\bar{t}_{accI} = 3,6 \text{ ns} + 3 \text{ ns} = 6,6 \text{ ns}$$

$$\bar{t}_{accD} = 3,6 \text{ ns} + 5,48 \text{ ns} = 9,08 \text{ ns}$$